REMARKS

Claims 30-66 and 105-124 are pending. Applicant has amended claim 30 to more clearly define the subject matter encompassed by Applicant's invention, and claim 51 to correct a minor typographical error.

The Examiner has rejected claims 30-66 and 105-124 under 35 U.S.C. § 112, first paragraph, as failing to comply with the written description requirement, and under 35 U.S.C. § 112, second paragraph, as being indefinite, asserting that the term "productive" has not been adequately defined. Applicant respectfully disagrees.

The term "productive" is used to describe a state of executing non-idle instructions. The specification defines this term in several locations. First at page 7, line 20, "the task is blocked from further productive use of the processor...rather than having the task continue to execute idle instructions (e.g., instructions looping checking for an event to occur)." Then at page 8, line 10, "productive work (e.g., non-idle instructions)." Therefore, Applicant submits that the term "productive" as used in the claims is supported by the specification and requests that this rejection be withdrawn.

The Examiner has rejected claims 30-66 and 105-124 under 35 U.S.C. § 103(a) as unpatentable over Borkenhagen et al (U.S.P. No. 6,567,839, hereinafter Borkenhagen). Applicant respectfully disagrees.

Borkenhagen describes a method for providing thread switching control in a multithreaded processor. In the system described by Borkenhagen, the multithreaded processor determines when to switch control from one thread to another thread rather than the thread notifying the processor when it wants to give up control. The switch may occur based on a latency event (e.g., a timer expires) or other events (e.g., the thread makes a system call) (col. 6, line 23-51, col. 7, line 22-30). The multithreaded processor will switch control to another thread without the active thread's involvement (col 6, line 28-29).

In contrast, Applicant's technology allows clients in a computer system to decide when to relinquish control of a resource to a server by providing the server with a notification, and may request that the server return that resource upon the occurrence of

an event. This differs significantly from the system in Borkenhagen where the processor decides when to assign and unassign the processor resource. Applicant's technique solves a problem in an environment where an operating system (or executive) does not have knowledge as to whether a task is in fact waiting for an event to occur. For example, a task may issue a memory read request and need to wait until the request completes before proceeding (e.g., a read request that needs to wait until another client writes to the memory request before it can be completed). In such a case, the task may need to loop checking for completion. Applicant's technology allows the task in this example to notify the operating system that it cannot use the processor resource until the memory read completes, which then allows the operating system to assign the processor resource to another task that can use the processor resource until the first task's memory read completes.

All of Applicant's independent claims recite these distinguishing characteristics. Claim 30 recites "receiving notification from the client assigned to the resource that the client is waiting for an occurrence of an event." Claim 41 recites "determining whether the client cannot use the resource until an event occurs...the client determines that it cannot use the resource." Claim 51 recites "receiving notification from the task assigned to the processor resource that the task is waiting for an occurrence of an event." Claim 59 recites "determining whether the task cannot use the processor resource until an event occurs...the task determines that it cannot use the processor resource." Claim 112 recites "means for notifying an operating system that the task cannot use the processor when the task determines that it cannot use the resource." In addition, the claims recite a unique combination of elements that are neither taught nor suggested by Borkenhagen.

Applicant respectfully submits that Borkenhagen fails to contain all of the elements of Applicant's claimed invention, and therefore does not render Applicant's invention unpatentable either individually, or in combination with the alleged skill in the art noted by the Examiner concerning the use of multithreaded technology in a client-server environment. Moreover, Applicant respectfully submits that the claims recite a unique combination of elements that is not obvious. Therefore, Applicant requests that the rejection under 35 U.S.C. § 103(a) be withdrawn.

Based upon these remarks, Applicant respectfully requests reconsideration of this application and its early allowance. If the Examiner has any questions or believes a telephone conference would expedite prosecution of this application, the Examiner is encouraged to call the undersigned at (206) 359-8548.

Respectfully submitted,
Perkins Coie LLP

Date: January 11, 2005

Maurice J. Pirio
Registration No. 33,273

Correspondence Address:
Customer No. 25096
Perkins Coie LLP
P.O. Box 1247
Seattle, Washington 98111-1247
(206) 359-8000